

# Работа с файловой системой

Таблица 12.1 – Свойства класса FileSystemInfo

<i>Свойство</i>	<i>Описание</i>
<b>Attributes</b>	Позволяет получить или установить атрибуты для данного объекта файловой системы. Для этого свойства используются значения и перечисления FileAttributes (изложено позже – в таблице 12.3).
<b>CreationTime</b>	Позволяет получить или установить время создания объекта файловой системы.
<b>Exists</b>	Может быть использовано для того, чтобы определить, существует ли данный объект файловой системы.
<b>Extension</b>	Позволяет получить расширение для файла.
<b>FullName</b>	Возвращает имя файла или каталога с указанием пути к нему в файловой системе.
<b>LastAccessTime</b>	Позволяет получить или установить время последнего обращения к объекту файловой системы.
<b>LastWriteTime</b>	Позволяет получить или установить время последнего внесения изменений в объект файловой системы.
<b>Name</b>	Возвращает имя указанного файла. Это свойство доступно только для чтения. Для каталогов возвращает имя последнего каталога в иерархии, если это возможно. Если нет, возвращает полностью определенное имя.

Таблица 12.2 – Наиболее важные члены класса DirectoryInfo

<i>Член</i>	<i>Описание</i>
Create() CreateSubDirectory()	Создают каталог (или подкаталог) по указанному пути в файловой системе.
Delete()	Удаляет пустой каталог.
GetDirectories()	Позволяет получить доступ к подкаталогам текущего каталога (в виде массива объектов DirectoryInfo).
GetFiles()	Позволяет получить доступ к файлам текущего каталога (в виде массива объектов FileInfo).
MoveTo()	Перемещает каталог и все его содержимое на новый адрес в файловой системе.
Parent	Возвращает родительский каталог в иерархии файловой системы.

Таблица 12.3 – Возможные значения свойства Attributes

<i>Значение</i>	<i>Описание</i>
<b>Archive</b>	Этот атрибут используется приложениями при проведении резервного копирования, а в некоторых случаях – удаления старых файлов.
<b>Compressed</b>	Определяет, что файл является сжатым.
<b>Directory</b>	Определяет, что объект файловой системы является каталогом.
<b>Encrypted</b>	Определяет, что файл является зашифрованным.
<b>Hidden</b>	Определяет, что файл является скрытым (такой файл не будет выводиться при обычном просмотре каталога).
<b>Normal</b>	Определяет, что файл находится в обычном состоянии и для него установлены любые другие атрибуты. Этот атрибут не может использоваться с другими атрибутами.
<b>Offline</b>	Файл (расположенный на сервере) хэширован в хранилище off-line на клиентском компьютере. Возможно, что данные этого файла уже устарели.
<b>ReadOnly</b>	Файл доступен только для чтения.
<b>System</b>	Файл является системным (то есть файл является частью операционной системы или используется исключительно операционной системой).

Таблица 12.4 – Основные методы класса Directory

<i>Метод</i>	<i>Описание</i>
<code>CreateDirectory()</code>	Создает папку с указанным именем.
<code>Exists()</code>	Проверяет существование папки с указанным именем.
<code>Delete()</code>	Удаляет папку с указанным именем.
<code>Move()</code>	Перемещает или переименовывает папку с указанным именем.
<code>GetLogicalDrives()</code>	Отображает имена всех дисков на данном компьютере.
<code>GetCurrentDirectory()</code>	Отображает имя текущего каталога

**Замечание.**

Удаление каталога возможно только тогда, когда он **пуст**.

Таблица 12.5 – Члены класса `FileInfo`

<i>Член</i>	<i>Описание</i>
<code>AppendText()</code>	Создает объект <code>StreamWriter</code> для добавления текста к файлу.
<code>CopyTo()</code>	Копирует уже существующий файл в новый файл.
<code>Create()</code>	Создает новый файл и возвращает объект <code>FileStream</code> для взаимодействия с этим файлом.
<code>CreateText()</code>	Создает объект <code>StreamWriter</code> для записи текстовых данных в новый файл.
<code>Delete()</code>	Удаляет файл, которому соответствует объект <code>FileInfo</code> .
<code>Directory</code>	Возвращает каталог, в котором расположен данный файл.
<code>DirectoryName</code>	Возвращает полный путь к данному файлу в файловой системе.
<code>Length</code>	Возвращает размер файла.
<code>MoveTo()</code>	Перемещает файл в указанное пользователем место (этот метод позволяет одновременно переименовать данный файл).
<code>Name</code>	Позволяет получить имя файла.
<code>Open()</code>	Открывает файл с указанными пользователем правами доступа на чтение, запись или совместное использование с другими пользователями.
<code>OpenRead()</code>	Создает объект <code>FileStream</code> , доступный только для чтения.
<code>OpenText()</code>	Создает объект <code>StreamReader</code> (о нем также будет рассказано ниже), который позволяет считывать информацию из существующего текстового файла.
<code>OpenWrite()</code>	Создает объект <code>FileStream</code> , доступный для чтения и записи.

Таблица 12.6 – Значения перечисления **FileShare**

<i>Значение</i>	<i>Описание</i>
<b>None</b>	Совместное использование открытого файла запрещено. На любой запрос на открытие данного файла будет возвращено сообщение об ошибке.
<b>Read</b>	Позволяет открывать файл одновременно и другим пользователям, но только на чтение. Если этот флаг не установлен, на любые запросы на открытие данного файла на чтение будет возвращаться сообщение об ошибке.
<b>ReadWrite</b>	Позволяет открывать файл одновременно и другим пользователям на чтение и запись.
<b>Write</b>	Позволяет открывать файл на запись одновременно нескольким пользователям.